Christoph von Braun

CS 4243 Design Document­­­­

11/9/15

DRATS stands for Drone Recognition And Tracking Suite. It is a system that is designed to identify a small quadcopter flying around a room and plot its coordinates in a 3D space. It operates using four cameras placed around a room, whose feeds are analyzed using an algorithm called SURF to detect the drone. The four individual matches are then combined using an algorithm of my own design to place that drone in an x-y-z coordinate space in a virtual model of the room. That information is then used in conjunction with two others’ projects to determine if the drone is behaving as it should, and hot swap the pilot program to a previously known state.

This program is designed to operate autonomously; therefore, there are no users, strictly speaking. An administrator of the suite will be able to view a graphical representation of where the program has detected the drone in each camera feed, as well as some historical data about where the drone used to be at some time in the past. The administrator will have no control over the program’s operation, however. It will accept no inputs of any kind.

The program will operate as follows: each camera feed will be individually running the SURF algorithm on a separate thread. This algorithm will be comparing patterns on still photos of the drone, and attempting to match them to the patterns seen by the camera. If enough match, it will consider that sufficient to claim that a drone has been found. SURF is called via OpenCV, which is an open source computer vision library. After detection, the center coordinate of the drone is taken, and sent back to the main program component, which should now have a set of 2D coordinates for each camera, corresponding to the drone’s relative location on that camera’s viewport. The coordinates are all run through an algorithm that takes into account the cameras’ relative positions to one another, and computes one set of x-y-z coordinates for the identified drone.

From the code standpoint, there will be three major functional components: main, SURF, and the 2D to 3D modeler. Main starts everything off, handles threading, initiates the components, and handles the communication between the components. Main is told the number of cameras that are present, and dynamically starts the individual threads and the SURF algorithm based on that information. The cameras are required to be in known positions, as their relative locations need to be hardcoded into the system. There is no calibration present currently, so they cannot move once the program has been started either. Another functional requirement is that there must only be one drone present in the monitored area. In DRATS’ current form, two identical drones will completely break the system. This is something that might be addressed in later builds. As SURF operates on patterns of the object itself, there is no need for there to be constraints on what is in the monitored area, such as movement, certain colors, and so forth.

The most prominent non-functional requirement of the project is that it should have some sort of user interface that displays past and present location data about the drone. The current working design is that it will show a green line on each camera, signifying where it has been over the last 30 seconds, and a green dot signifying where it currently is. As mentioned before, there is no user input required of the program, so this interface is simply for informational purposes only, and hence is not necessary for the system to function correctly.

